AD-A240 945

2

ANNUAL REPORT

VOLUME 3

PART 1

TASK 3: SPECIAL STUDIES

REPORT NO. AR-0142-91-002

September 27, 1991

GUIDANCE, NAVIGATION AND CONTROL

DIGITAL EMULATION TECHNOLOGY LABORATORY

Contract No. DASG60-89-C-0142

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332-0540

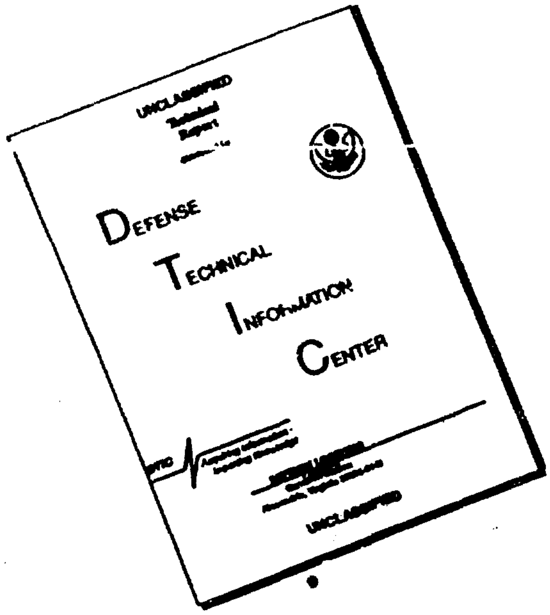
Contract Data Requirements List Item A005

Period Covered: FY 91

Type Report: Annual

91-11308

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

DISCLAIMER

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

DISTRIBUTION CONTROL

(1) This material is approved for public release unlimited distribution.

(2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227–7013, October 1988.

ANNUAL REPORT

VOLUME 3

PART 1

TASK 3: SPECIAL STUDIES

September 27, 1991

Authors

Cecil O. Alford, Richard M. Pitts and Philip R. Bingham

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332-0540

Eugene L. Sanders
USASDC
Contract Monitor

Cecil O. Alford
Georgia Tech
Project Director

TABLE OF CONTENTS

List of Figures

TASK 3: SPECIAL STUDIES

1. INTRODUCTION

The task objectives under the contract are to develop GN&C Processing technology, to provide support to LATS and AHAT, to develop software for the PFP and flight processor and to develop simulations and test software to evaluate GN&C processors. These tasks, and their interconnections are shown in Figure 1.1.

Software development is being done for the PFP and the GN&C prototype. As shown in Figure 1.2 there are seven processors with a number of software modules for each. Many of these modules have been demonstrated. The most pressing is the development of a validated Ada compiler for the GT-EP processor. This is under contract to Basis Technology Corporation and Irvine Compiler Corporation.

The PFP is used as an emulation tool for KEW interceptor systems. Figure 1.3 shows the status of current KEW interceptor emulations. EXOSIM, explained in Section 4, has been demonstrated, and continues to be upgraded. A small amount of work has been done on Army Leap, but no official Leap simulation is at DETL. GBI and E^2I are in the planning stage and no simulation exists at DETL.

Special purpose software, shown in Figure 1.4, consists of Signal Processing benchmarks, KEW Flight Software benchmarks, Block Diagram Editing Tools, Scene Generation Tools, and Parallel Programming Tools. The Signal Processing Benchmark has been completed and two reports published. The first version was in Fortran [1]; the second version was in Ada [2]. The other benchmarks and tools are under development.

VLSI chip status is shown in Figure 1.5. Most of the designs have completed fabrication and are in testing as individual chips or more advanced testing in a module. All eleven AHAT VLSI designs have been delivered. Four new VLSI designs are underway. These are needed to supplement the current chip set and to extend capability. In addition, new features are planned for some of the original 13 chips.

Modules for the GN&C prototype are shown in Figure 1.6. The design for most of the modules is complete. Some boards have been fabricated and are in testing. Two of the boards were demonstrated, but are being revised. Only the status of the revised boards is given.

Task 3 was set up with two objectives. The first was to establish interfaces between Georgia Tech and other programs such as LATS, LETS, AHAT and KDEC. Each of these programs requires specific interactions and data exchange with Georgia Tech. Some of these will eventually require an interface specification to define hardware and software boundaries between Georgia Tech and the other programs.

The second objective of Task 3 is to resolve issues which were not anticipated in the original contract, but which need resolution for the contract to move forward. These

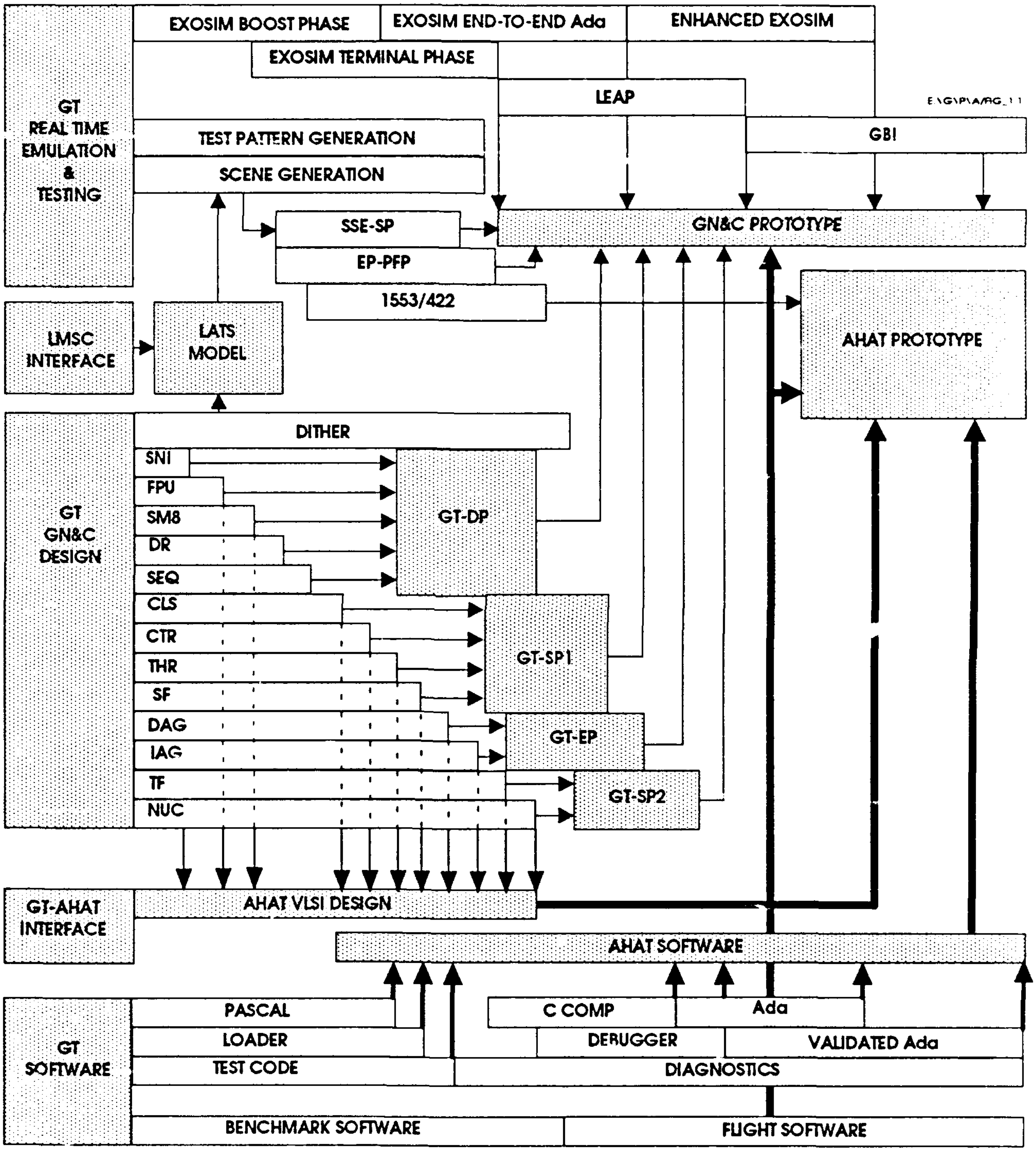


Figure 1.1 DETL Programmatic Tasks

E:\G\P\A/FIG_1.2

SOFTWARE DEVELOPEMENT							
CODE	FPX	FPX	DP	EP	386	860	486
PASCAL COMPILER	D	D	D	D	D	P	P
FORTRAN COMPILER	D	D		P	D	P	P
C COMPILER	D	D	N/A	C	D	P	P
DEBUGGER	N/A	N/A	N/A	C	N/A	P	N/A
Ada COMPILER	D	D	N/A	C	D	P	P
LOADER	D	D	D	P	D	P	P
HOST UTILITIES	D	D	D	P	D	P	P
MATH LIBRARIES	D	D	D	C	D	P	P
RUN TIME KERNEL	D	N/A	N/A	P	D	P	P
KEW APPLICATION SW	D	D	D	P	D	P	P
Ada VALIDATION	N/A	N/A	N/A	C	P	P	P

P: In Progress at DETL
T: In Test at DETL
C: Under Contract to Irvine Compiler Corporation and Basis Technology Corporation
D: Demonstrated
N/A: Not Applicable

Figure 1.2 GN&C and PFP Software Development

E:\G\P\A/FIG_1.3

KEW EMULATION				
MODULE	SINGLE PROCESSOR	MULTI PROCESSOR	Ada	R/T
Exosim Boost Phase	D	D,24	D	D
Exosim Terminal Phase	D	D,13		D
Leap				
GBI End-to-End				
E^2I End-to-End				

P: In Progress at DETL
T: In Test at DETL
D: Demonstrated On PFP
D,XX: Demonstrated On PFP Using XX Processors

Figure 1.3 KEW Interceptor Emulation Status

E\G\P\A/FIG_1.4

SPECIAL PURPOSE SOFTWARE				
MODULE	FORTRAN	C	Ada	STATUS
SP Benchmark	D		D	U
128 x 128 Multi Target SP	D			
Parallel Tools	D			
Flight SW Benchmark				
Block Diagram Editor				

D: Demonstrated at DETL
U: User Manual Exists

Figure 1.4 Special Purpose Software Development Status

E\G\P\A/FIG_1.5

VLSI DESIGN					
VLSI CHIP	DESIGN	DES VER	FAB	STATUS	AHAT
NUC	C	C	C	H	DEL
THR	C	C	C	T	DEL
TF	C	C	C	T	DEL
SF	C	C	C	T	DEL
CLS	C	C	C	D	DEL
CTR	C	C	C	D	DEL
IAG	C	C	C	T	DEL
DAG	C	C	C	T	DEL
SNI	C	C	C	D	DEL
FPU	C	C	C	D	DEL
SM8	C	C	C	D	DEL
SEQ	C	C	C	D	
DR	C	C	C	D	
DIT	P				
SENSOR FUSION	S				
NEURAL NET	S				
DPFPU	S				

C: Complete
P: In Progress at DETL
S: In Specification at DETL
T: In Test at DETL
H: In Test at HP
D: Demonstrated In Processor Module
DEL: Delivered Design Tape

Figure 1.5 VLSI Chip Set Design Status

E\G\P\A/FIG_1.6

GN&C PROTOTYPE			
Module	DESIGN	FABRICATION	STATUS
Host Interface	C	C	D
SSE Interface	P		
NUC/TF	P		
THR/SF	C	C	T
CLS/CTR	C	C	D
DP	C	C	D
EP	C	C	T
1553 Interface	C	P	
SM8/RS232	S		

C: Complete
P: In Progress at DETL
S: In Specification at DETL
T: In Test at DETL
D: Demonstrated

Figure 1.6 GN&C Processor Prototype Development

issues are sometimes quite difficult and beyond the resources of the contract. However, each issue is at least defined and evaluated in importance. If further work is justified it is left open for USASDC assignment.

The specific tasks under study and schedules for each are shown in Figure 1.7.

Figure 1.7. Georgia Tech Special Studies Development Schedule

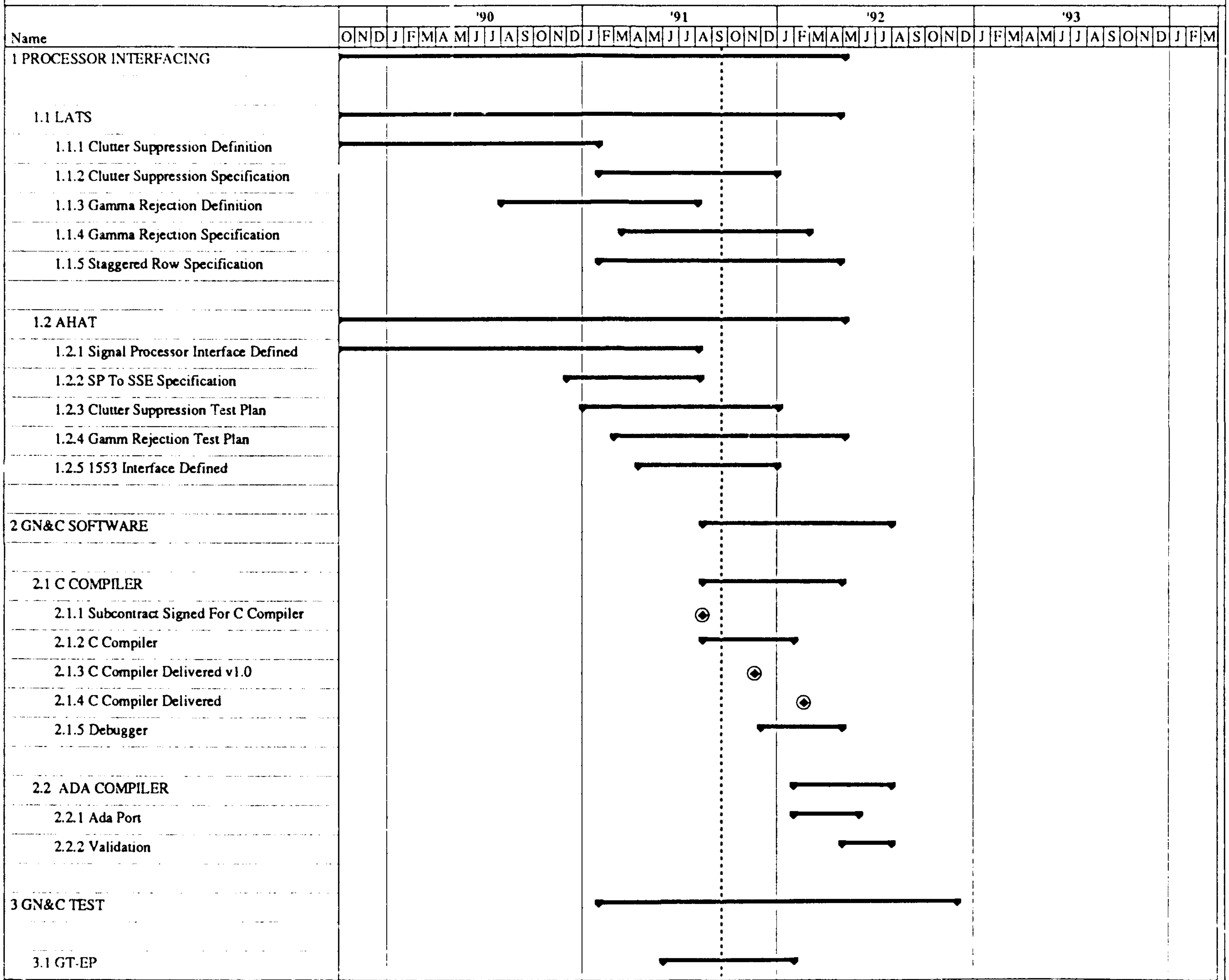


Figure 1.7. Georgia Tech Special Studies Development Schedule

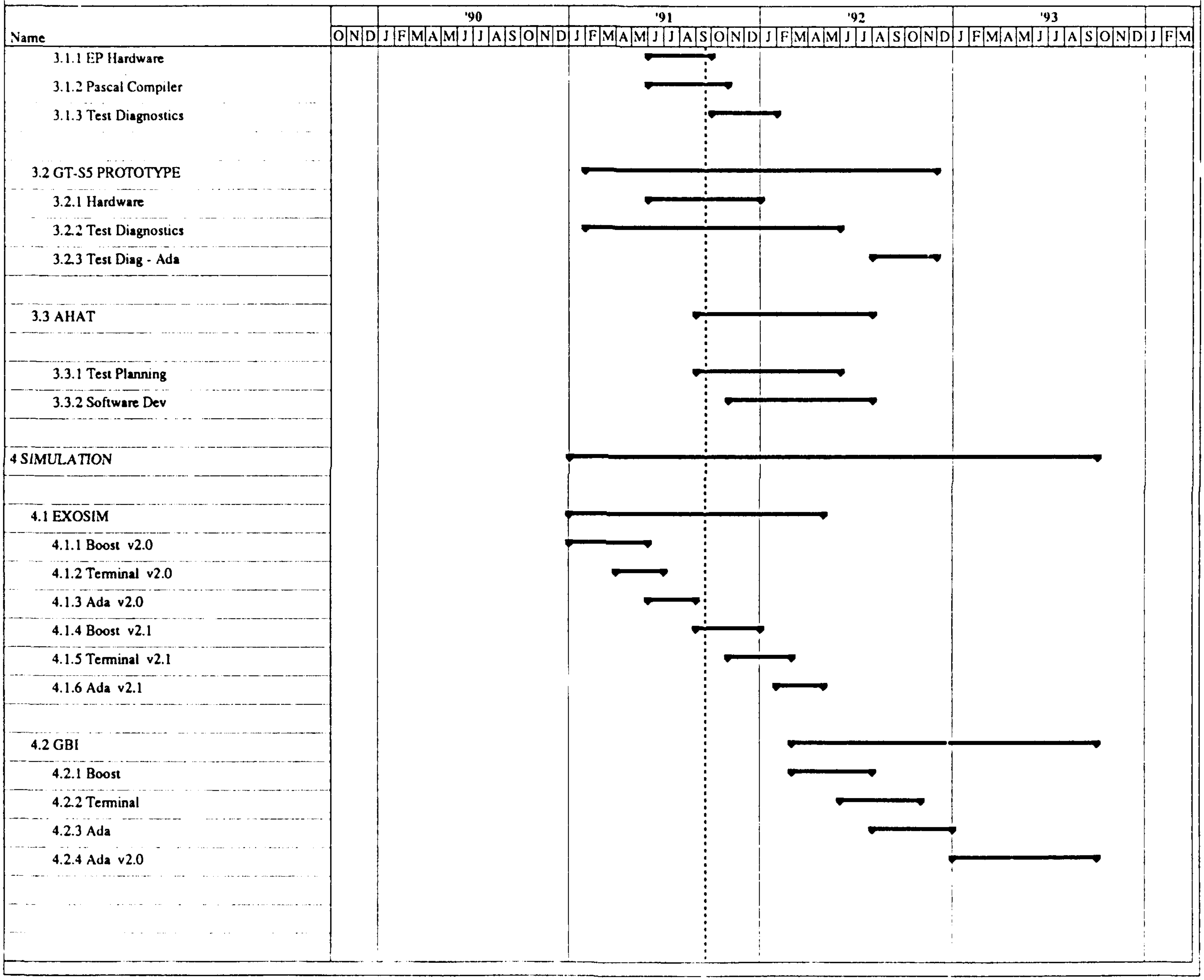
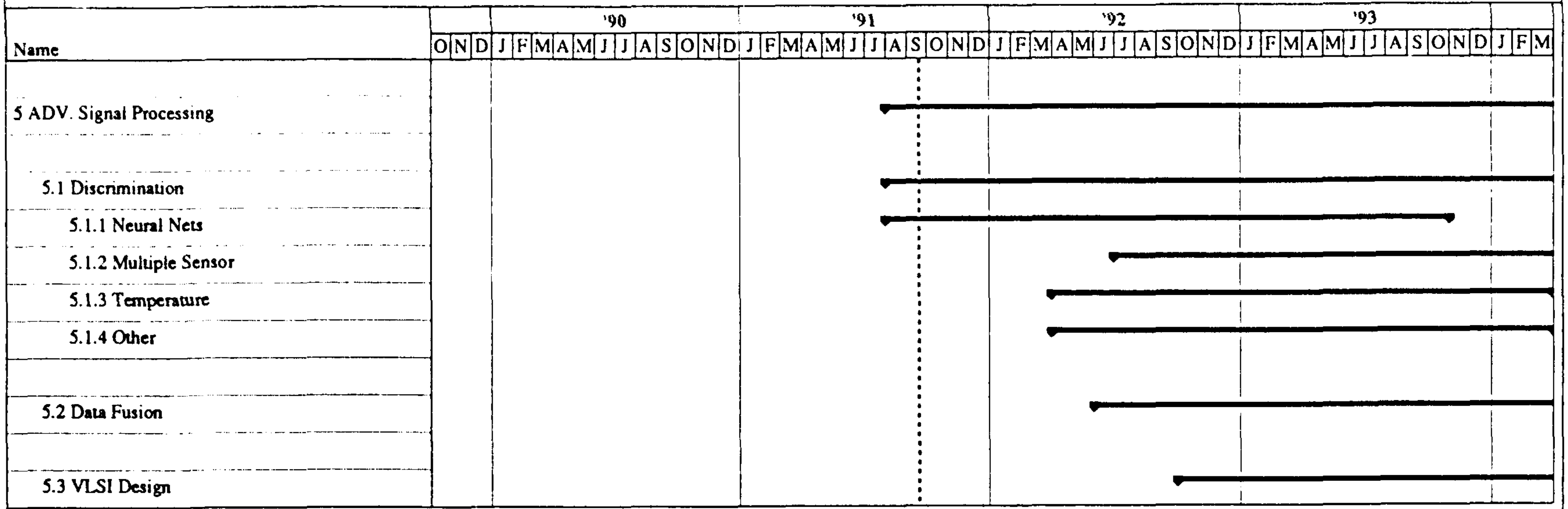


Figure 1.7. Georgia Tech Special Studies Development Schedule



2. CONTRACT INTERFACES

Interfaces have been established with all four programs which merge with Georgia Tech. These are AHAT, LATS, Jaycor and KDEC. A point of contact has been established at Georgia Tech for each program. A summary of each follows.

2.1 AHAT

Joseph Chamdani is the point of contact for VLSI design and GN&C Processor issues. Andy Register is the Georgia Tech interface for all AHAT meetings. This interface is working smoothly. Georgia Tech VLSI design tapes have been delivered to Harris on a regular basis. Questions that arise are resolved quickly if the information is at Georgia Tech. The specific interface between the AHAT processor and the Georgia Tech test facility is being addressed. Two issues are outstanding. The first is the 1553 interface for instruction loading and editing. The second is the data port for FPA pixel information. These two have been defined and Georgia Tech has begun interface designs.

Software is another key AHAT issue. This is addressed in Volume 4. Testing the AHAT processor requires the development of special patterns to verify functional operation and timing methods to verify speed of operation. These are being addressed as a part of the Seeker/Scene Emulator in Volume 2.

2.2 LATS

The LATS seeker will not be placed in the LETS facility and there are no plans to connect it to the PFP. Previous plans for these interfaces have been dropped.

The interface issues with LATS are: (1) the connection to the AHAT processor or the GT Prototype, (2) the staggered row Focal Plane Array, (3) the Gamma Suppression methodology and (4) the Dithering technique for spatial filtering. Andy Register is the point of contact with LATS. Volume 4 addresses issues (2) and (4). Issue (3) is currently being implemented in analog technology on the FPA. Georgia Tech has a preliminary design ready, but has placed all further work on hold pending the results of the analog approach. The interface is addressed in Volume 4.

2.3 JAYCOR

Randy Abler is the primary interface between Georgia Tech and the AHAT Test Article team. Jaycor is one of several contractors involved in this program. The primary responsibility for Georgia Tech is the delivery of the VLSI chip designs to Harris and the development of appropriate software to support and test the AHAT processor. Georgia Tech has no contract for specific work with Jaycor. It is possible that some test issues will require new work for Georgia Tech.

2.4 KDEC

A PFP was delivered and installed in the KDEC facility at USASDC. Richard Pitts is the point-of-contact. Georgia Tech has provided PFP programming classes to support KDEC. Two manuals have been written to assist users in programming the PFP. Several simulations have been delivered to KDEC for parallel execution.

2.5 Other

Interfaces with Draper Laboratories, LETS, and KHILS are not being developed. As these entities become important and require special attention, new interfaces will be developed.

Interfaces with LEAP, E^2I, GBI, ERINT and other program offices are being pursued. There is no current activity other than planning and demonstrations of the DETL capability.

2.6 Working Groups

To support the interfaces and to ensure that all participants have a forum for presenting requirements, specifications and problems, a set of four working groups (WG) has been established. Each WG is directed at a specific technology area. The four groups and their leaders are:

(a) Parallel Simulation Technology Working Group - Tom Collins (404)894-2509. This group will focus on parallel simulation techniques and tools, simulation languages, and parallel simulations such as EXOSIM and LEAP.

(b) Seeker/Scene Emulation Technology Working Group - Andrew Henshaw (404)894-2521. This group will focus on scene generation techniques, seeker modeling, benchmark test and evaluation data for GN&C processors, and emulation accuracy assessment.

(c) GN&C Processor Working Group - Andy Register (404)894-3812. This group will focus on processor architectures, VLSI design tools, VLSI fabrication, packaging and benchmarks for processor evaluation and testing.

(d) Parallel Ada Software Working Group - Randy Abler (404) 894-2531. This group will focus on development tools for embedded Ada software, parallel implementation techniques, test and evaluation of embedded Ada software, and benchmark test programs for parallel simulation and GN&C processors.

These groups have not been active this year, but are still formed and available. If the need arises they can be used to assist the transfer of information to other units.

3. TECHNICAL ISSUES

3.1 LATS Seeker

3.1.1 Dithering

The LATS Seeker requires a special function to remove spatial noise. This function is based on a dither motion of the FPA and requires computations during explicit modes termed scan and stare. In addition to the filtering operation, the function also performs time dependent integration, converting the pixel stream from 1 Khz to 100 hertz.

Georgia Tech began a VLSI design to support the mathematical function of dithering. During the design two issues were raised. The first issue is the explicit function to be implemented. There is still some question on the definition, and even on the form. The second question concerns the interface signals required. Since this chip will be the first digital chip in the signal processing chain, it is desirable to put all the required interface signals on this chip. Again, these signals are not well defined.

Georgia Tech will continue to work with LATS to define and design the dithering chip such that it will provide the required functions.

3.1.2 Delayed Gamma Model

The rejection of gamma spikes in the signal processor presents a significant problem. Data rates are required to be quite high (10 khz) which requires a large number of A/D converters. Processing, although quite simple, requires a large number of chips to support the high data rate. An effort was made to develop a reasonable model to test various gamma rejection methods. This work is reported in Volume 5. Ultimately an analog scheme is preferred since it would eliminate chips and reduce the package size and weight. It is too early to tell whether analog techniques are sufficient, but the assumption is being made that they will eventually do the job. In case they do not, a digital plan is available and can be implemented.

One of the outputs from the investigation was a testing procedure. Georgia Tech has a preliminary design for a gamma injection circuit which could be used to test signal processing hardware with digital gamma rejection. Since the input is known, it is possible to get an exact measure on any gamma rejection hardware. This could prove useful, but only for digital methods, since analog schemes require a different injection technique. Nothing further will be pursued in the gamma rejection or testing area unless the analog technique fails.

3.1.3 Staggered Row FPA

The Georgia Institute of Technology (Ga. Tech) is developing a set of application specific integrated circuits (ASIC) to perform the signal processing functions required in

an exo/endo atmospheric interceptor (GBI, GBI-X, E^2I, LEAP, etc.). The ASIC signal processor (SP) was designed assuming a non-staggered or uniform 128x128 staring FPA. Some researchers in the seeker community are studying the use of staggered pixel geometries for the FPA. The staggered FPAs offer improved signal to noise ratio (SNR) for small objects. This is important for long range acquisition.

A report was written to examine the methods and trade-offs for interfacing a staggered FPA with the Ga. Tech SP [3]. The examination begins by attempting to derive a comparable figure-of-merit for any geometry. This figure-of-merit is calculated for the geometries known to be under consideration for the FPA. A variety of mapping functions are presented and compared. Most of these mappings must be applied before the pixels enter the SP. One promising map is a post-processed map that acts on the object centroids. This map is attractive because it can be computed in the object processing (OP) portion of the seeker.

Two preliminary analyses are performed. The first is a worst case analysis based on an object size approximately equal to one picture element (pixel). The second is a more realistic analysis based on blur-spot assumptions.

3.2 Discrimination Techniques

3.2.1 Neural Network

Georgia Tech initiated a study of Discrimination Techniques based on Neural Networks to identify methods for feature extraction. The initial approach uses higher order moments of defined shapes as the measure for classification. An agreement was reached with Lucid Incorporated to use their software package, Plexi, as the design analysis tool. The software was installed on a SUN workstation with Joe Pendergrass as the primary engineer.

Georgia Tech has defined a simple neural network, specified some shapes, trained the network, and evaluated the ability of the network to discriminate based on moments. This work was used in a technology demonstration July 18, 1991. All of this is preliminary and is being extended to more complicated environments and targets.

3.2.2 Temperature

Georgia Tech began a study to determine the feasibility of using one color as a discrimination measure. There are some reports which indicate the accuracy requirements are too severe for this to be practical. However, this work is the front-end for a two color scheme which will be used independently and in combination with the Neural Network approach. This work is preliminary and no definitive results are available.

3.2.3 Multiple Sensors

Georgia Tech is planning on supporting two color sensors and one visible sensor in the prototype processor. This requires the design of a sensor fusion element to blend the data. In the architecture, the first method would use three object processors, to process each data stream separately. After centroids are determined these will have to be treated separately and in combination. This is on-going work which is to be done in FY 92.

3.3 Parallel EXOSIM

A parallel simulation for an EXO-Atmospheric KEW interceptor has been developed by Coleman Research Corporation. Converting the Fortran code for PFP execution has been a primary objective of Task 1. The unusual problems encountered by the Task 1 group and by Dynetics Inc., necessitated additional effort and approaches. Volume 3, Section 4 of the FY 90 Annual Report presented a summary of a conversion methodology developed under Task 3. Section 5 of that report presented the parallel implementation of the V1.0 boost phase.

3.3.1 Boost Phase

Section 4 of this report continues the work on the Boost Phase of EXOSIM which was discussed in Volume 3, Section 5 of the FY 90 Annual Report. The parallel implementation has now been extended to Ada in a parallel, real-time implementation.

3.3.2 Midcourse/Terminal Phase

The Midcourse/Terminal Phase is a classified program in V2.0 of EXOSIM. This part of the flyout was removed from the end-to-end simulation to simplify the parallel implementation. Section 4 of this report describes the parallel simulation based on Fortran source code and C.

3.4 Benchmarks

3.4.1 Signal Processing Benchmark

Two benchmarks have been generated for signal processing. These were reported as STR-0142-90-008 [1] and STR-0142-91-0142 [2]. The first is a Fortran benchmark and the second an Ada benchmark. Both are derived from a set of OCCAM subroutines which were used to design and test the Georgia Tech VLSI Signal Processing chips.

These documents describe a set of signal-processing algorithms, as implemented by the Computer Engineering Research Laboratory at Georgia Tech. The routines are presented as a representative collection of operations for processing Infrared Focal-Plane Array signals.

For the purposes of testing and dissemination, each algorithm is presented as a stand-alone Ada (Fortran) program. These programs are based upon a core harness routine which supports input/output of a common data format (Georgia Tech Algorithm

Evaluation Data Format - described in the Harness section). The modular implementations offer several benefits:

- simplification of the generation of test vectors for the verification of alternate implementations
- capability for testing various algorithm combinations, without re-compilation
- support for multiple language and/or processor-platform implementations

The harness program is the basis of the input/output methodology used by all of the routines. The code implements a simple Pass-Through module which reads a data stream, picking off the FPA pixel data, and writing the data onto an output data stream.

The Georgia Tech Algorithm Evaluation Data Format is a simple ASCII text representation of a data stream. The data stream has two major components - the Field Header and the Field Data. The harness of each module processes the data stream by reading each line and checking for Field headers which are relevant to that module. Any lines which are not relevant, or unrecognized, are immediately placed upon the output data stream. As soon as a relevant Field Header is recognized, the Field Data which follows is processed in a manner which is appropriate to that module and Field header. This scheme provides for the chaining of modules output-to-input, without either module requiring knowledge of all, or any, of the other module's data formats. In typical use, controls for many modules could be included in a single data stream; each module would only process data intended for it. Further details on these two benchmarks can be found in the two Special Technical Reports.

3.4.2 Simulation Benchmarks

Several parallel simulations have been developed and sent to KDEC under a separate contract. These simulations are typical of many KEW applications and some are versions of EXOSIM described in Section 4. Any of these simulations can be used as a parallel processing benchmark in a variety of ways. Newer versions are being added to this collection which stress the seeker and signal processing capabilities of simulation hardware. All benchmarks are provided with a tape and documentation for loading onto the PFP and executing. Further details are available in past reports to USASDC.

4. PARALLEL PROGRAMMING METHODOLOGY

4.1 Introduction

The Digital Emulation Technology Laboratory was built with the specific purpose of real-time, parallel emulation of Kinetic Energy weapons problems. In order to model these systems a parallel functional processing concept was utilized. This concept uses a methodology which preserves boundaries between physical blocks in the system in order to simplify design changes, component testing and performance evaluation.

The Seeker/Scene Emulator is capable of modeling both background and target characteristics and staring array seekers at the pixel level. The SSE can generate pixel information representative of 128x128 focal plane arrays at rates up to 100 frames per second. Most importantly, hundreds of objects can be modeled for each frame of data. This same device can be used to generate prescribed patterns of data, to evaluate and test signal and data processing hardware.

The combined PFP-SSE-GN&C processing structure has demonstrated real-time operation of the KEW EXOSIM engagement. Part of this demonstration was in Ada code. The remainder was in Fortran and is now being converted to Ada.

An Exoatomospheric Kinetic Energy Weapons Interceptor Simulation was initiated by BDM Corporation and completed by Coleman Research Corporation. The final report, EXOSIM V2.0, was delivered to the U.S. Army Strategic Defense Command, October 15, 1989 under contract number DASG60-88-C-0002, Task Assignment Number TAQ - 002 [6,7]. This simulation, written in FORTRAN, was developed to run on a serial computer such as a VAX. The objective at Georgia Tech is to convert the simulation to Ada and to restructure it in a parallel format for execution on the Georgia Tech Parallel Function Processor. A second objective is to preserve the functional nature of the simulation so that changes can be easily incorporated into the executing code.

An analysis of the serial code revealed several problem areas which make the conversion more difficult. These are:

1. Double Precision Variables.
2. Dependent Partitions.
3. Event driven code.
4. System States difficult to identify.
5. Inadequate descriptions of system models.

Two approaches were selected for the conversion. The first, and most direct, was to examine the code, extract the physics, and re-write the simulation. This was considered

to be a very long term effort and not necessary at this time. The second approach was initiated by Dynetics Inc. Dynetics removed the event driven structure, produced an "equivalent unclassified version", and identified partitions which could be run in parallel. Dynetics worked with EXOSIM V1.0 since V2.0 was not available when they began work. They were only able to complete the Boost phase before funds were exhausted. This code was delivered to Georgia Tech and is known as EXOSIM Boost2.

4.2 The EXOSIM Engagement

The EXOSIM engagement, shown in Figure 4.1, contains n interceptor with a staring array seeker and one or more objects in the threat. [6,7] Real-time emulation is possible without a GN&C processor if a simple seeker is modeled and only one or two objects are used. For a real staring array and multiple objects, the processing demands are too large for the processing elements normally used in the PFP. However, it should be emphasized that any processor can be configured for the PFP and it is possible to construct a signal processor to perform this function in real-time.

The EXOSIM engagement, shown in Figure 4.1, was developed as a serial Fortran program by Coleman Research Corporation. The system blocks are shown in Figure 4.2. The engagement was separated into a boost phase and a terminal phase for implementation. The development of each is discussed in the following sections.

4.3. Midcourse/Terminal Phase Simulation

EXOSIM 2.0 was originally a serial event driven Fortran simulation. Before splitting the program into parallel parts, several steps were taken to prepare the program for parallelization and to test the program for the possibility of a parallel implementation. After converting to a time driven simulation, EXOSIM was executed and a restart file was saved after the boost phase of the simulation. This gave the capability of starting the simulation after the boost phase and thus the subroutines that only control boost phase functions were removed from the simulation leaving a midcourse/terminal phase simulation. Figure 4.3 shows all of the remaining subroutines and the communication between the subroutines. In order to test the program for the possibility of a parallel implementation, delays were inserted into the serial program to simulate 5 parallel partitions. These partitions were made using a model in which all partitions calculate and then communicate during a single cycle of the program. Inspection of the KEW calculated miss distance indicated EXOSIM could be split into these 5 partitions and placed on the PFP.

All of the previous changes were made on a Microvax computer. Figure 4.4 shows the programming framework that was used in the development of a real-time version of the terminal phase of EXOSIM. The code on the Microvax was ported to the Intel RMX host connected to a PFP filled with 386 processors. Using the partitions that were tested on the microvax, the code was split into 5 partitions and executed on the PFP. This halved the simulation time. Using the calculated miss distance as a test of the validity of each new partition, a ten partition simulation was eventually obtained that operated 5 times slower than real time on the PFP using 386 processors. Since the entire

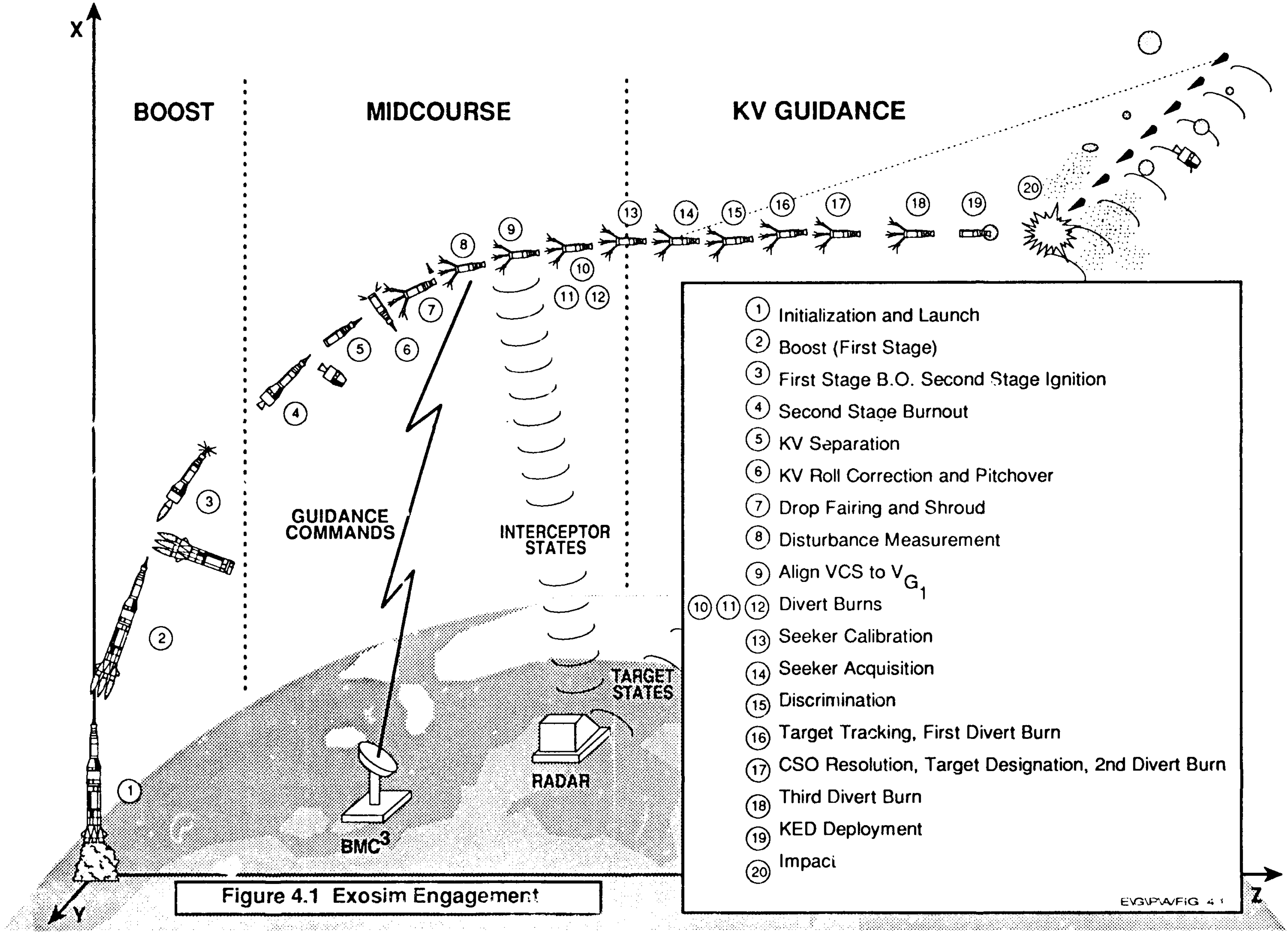


Figure 4.1 Exosim Engagement

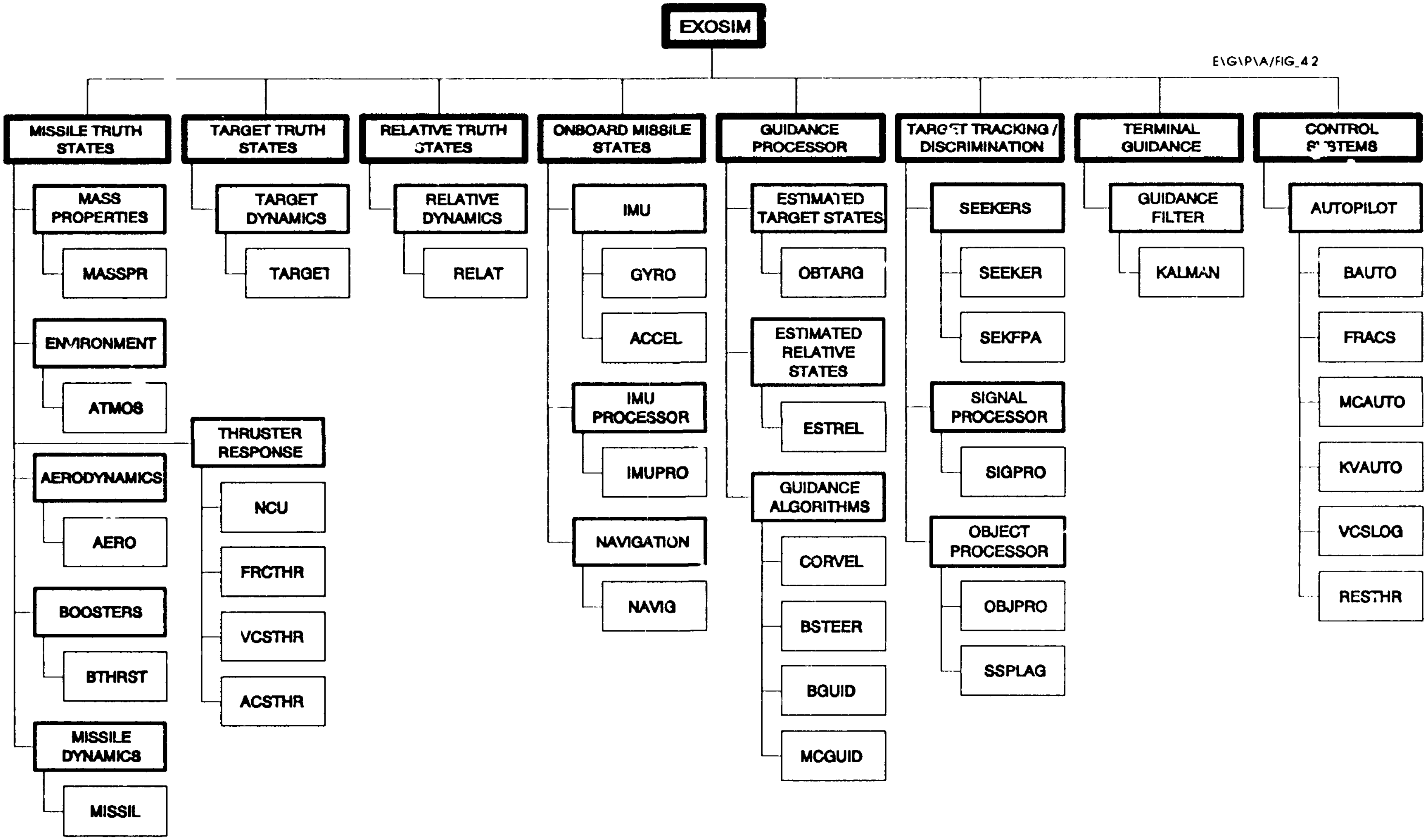


Figure 4.2 Functional System Blocks for Exosim Engagement

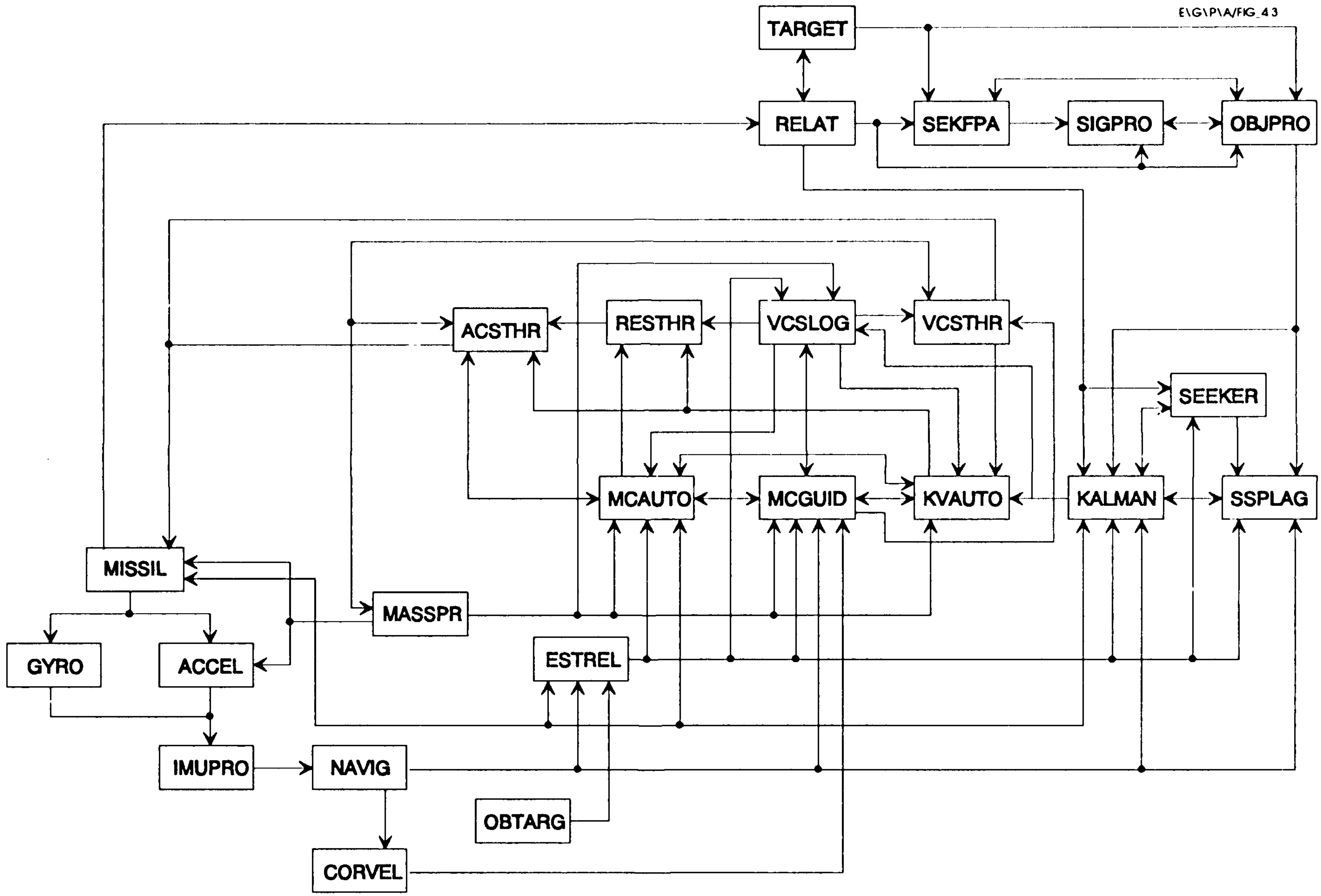


Figure 4.3 Exosim Midcourse/Terminal Phase Block Diagram

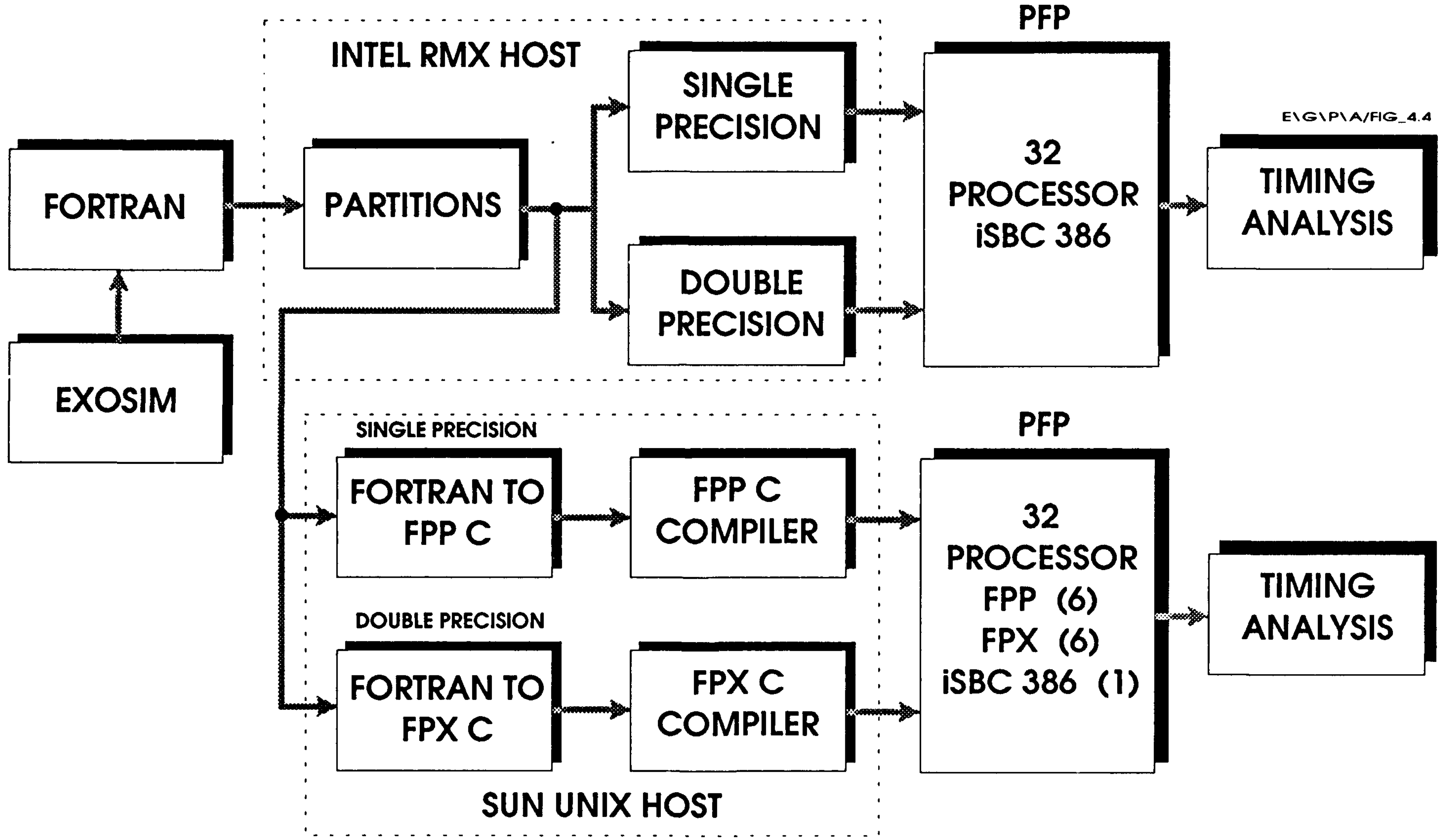


Figure 4.4 Programming Framework for Parallel Implementation of the Terminal Phase

simulation was written in double precision, the next step taken was a conversion to single precision. Partitions were converted to single precision one at a time until all partitions that could operate in single precision, without changing the miss distance, were converted. Converting to single precision brought the simulation to within 4.5 times real time.

The 10 partition FORTRAN version of EXOSIM was ported to the Sun host connected to a PFP containing FPP, FPX,and 386 processors. All of the partitions were converted from FORTRAN to C so that the partitions could be compiled for the FPP and FPX boards. Those partitions that were too large to fit in the FPP and FPX memory were compiled for the 386 processors. Using three different kinds of processors, the simulation ran 2 times slower than real time. The slowest parts of the simulation were the partitions on the 386 single board computers. Several more partitions were created on the Intel RMX host and brought over to be placed on the FPP and FPX processors which resulted in faster than real time execution.

The final version of EXOSIM contained 13 partitions executing on 3 different types of processors (6 FPP partitions, 6 FPX partitions, and one 386 partition). Figure 4.5 shows the subroutines contained in each of the 13 partitions. These subroutines can be traced back to those in the serial version shown in Figure 4.3. Two differences in subroutine names can be seen between the figures. In Figure 4.5 several subroutines are lumped together under the label AP. This indicates the autopilot subroutines and includes MCAUTO, KVAUTO, VCSLOG, and RESTHR. The subroutines with a "2" at the end of their name were created when a subroutine had to be split during partitioning. Also included in Figure 4.5 are code sizes for instructions and data for each processor, and the types of processor used to implement specific functions.

Finally, a graphics interface was developed on the Sun host that uses Sunview graphics windows to show the missile and target position in two different places. One plane shows the entire midcourse and terminal phase and the other plane zooms in and shows a close-up of the three diverts at the end of the simulation. All significant events are labeled along the flight path and several flight parameters (ie. altitude) are displayed in the window.

4.3.1 PFP Test Results

The criteria used to decide the validity of any changes to the simulation was the nearest miss calculation. If the nearest miss was within a certain limit, and all of the significant events (KV orientation, midcourse burns, divert burns, frame rate changes) occurred in the same sequence as the serial simulation the new partition was accepted. The 13 processor partition passed this criteria. In order to more thoroughly test the parallel version of EXOSIM changes were made in the starting position of the target and the change in the nearest miss distance given by the parallel version were compared to the change in nearest miss distance obtained by the serial version. The 13 partition parallel simulation also passed this test, behaving very similar to the serial version running on the Intel RMX machine.

E\G\P\A/FIG_4.5

Processor	Type	Instruction Bytes	Data Bytes	SIMULATION MODULE				
P0	FPP	25,020	2,848	INIT	MASSPR	MISSIL2	INTEG	
P1	FPX	48,480	3,384	INIT	IMUPRO	NAVIG	OBTARG	ESTRL2
P2	FPP	17,592	1,144	INIT	VCSTHR			
P3	FPX	48,684	3,336	INIT	RELAT	TARGET		
P4	FPX	45,732	3,932	INIT	GYRO			
P5	FPP	41,520	2,676	INIT	ACSTHR			
P6	FPP	7,308	524	INIT	MASSPR2			
P7	FPP	16,176	1,052	INIT	CORVEL			
P8	FPX	35,508	4,372	INIT	MISSIL			
P9	FPX	16,080	1,856	INIT	SEEKER			
P10	FPX	39,204	3,976	INIT	ACCEL			
P11	386	112,000	N/A	INIT	VCSTHR2	ESTRL	MCGUID	AP
P12	FPP	28,668	2,664	INIT	KALMAN			

Figure 4.5 Parallel Exosim: Terminal Phase Implementation

4.4 Boost Phase Simulation [4, 5]

Previous work described in Section 4.1, produced a real-time, parallel version of the boost phase. The problem now was to take this running model and convert the code from C to Ada.

The block diagram for the boost phase is shown in Figure 4.6. This structure is implemented in 24 processors or modules. As was true in the Terminal Phase, some blocks in Figure 4.6 were split. The code for each of the 24 processors was written in Fortran and translated to C as shown in Figure 4.7. The results were compared to the original Fortran version running on a single processor.

Each of the Fortran programs was translated to Ada, and the Irvine Compiler was used to produce C code as shown in Figure 4.8. The Fortran to C code was used as the starting point. One Fortran block would be replaced by an equivalent Ada block, followed by test execution. If the results were accurate the next Ada block would be loaded and tested. The end result contained 20 GT-FPP processors, 4 GT-FPX processors and no 386 processors.

The code produced by each methodology was examined with the results shown in Figures 4.9 and 4.10. In Figure 4.9 the ratio of Ada source code to Fortran source code is shown where Ada was produced by translating Fortran into Ada. The other curve in Figure 4.9 shows the ratio of C code generated by Ada to that generated by Fortran. The source code indicates 20% to 30% more Ada code, while the C code is only about 10% greater for Ada.

The object code ratio shown in Figure 4.10 shows almost equivalent sizes except for a few processors. The cases which have large Ada object code results from arrays which are compiled inefficiently by Ada. Execution time, however, is slightly faster for Ada than for Fortran. Both give faster than real-time performance.

4.5 Comparative Results

The EXOSIM engagement has been executed on several computers using the simple seeker and a single target. This represents the least complex engagement, but provides a uniform baseline for comparison. Relative execution times are shown in Figure 4.11 for several common processors or computers. Additional benchmarks are being executed and will be added to the chart upon completion. For this chart, the PFP was used as a stand-alone unit, without the SSE or GN&C processor. For real-time emulation of a flight engagement, both would be necessary. However, for testing software and most interceptor performance studies, the simple seeker-single target is sufficient.

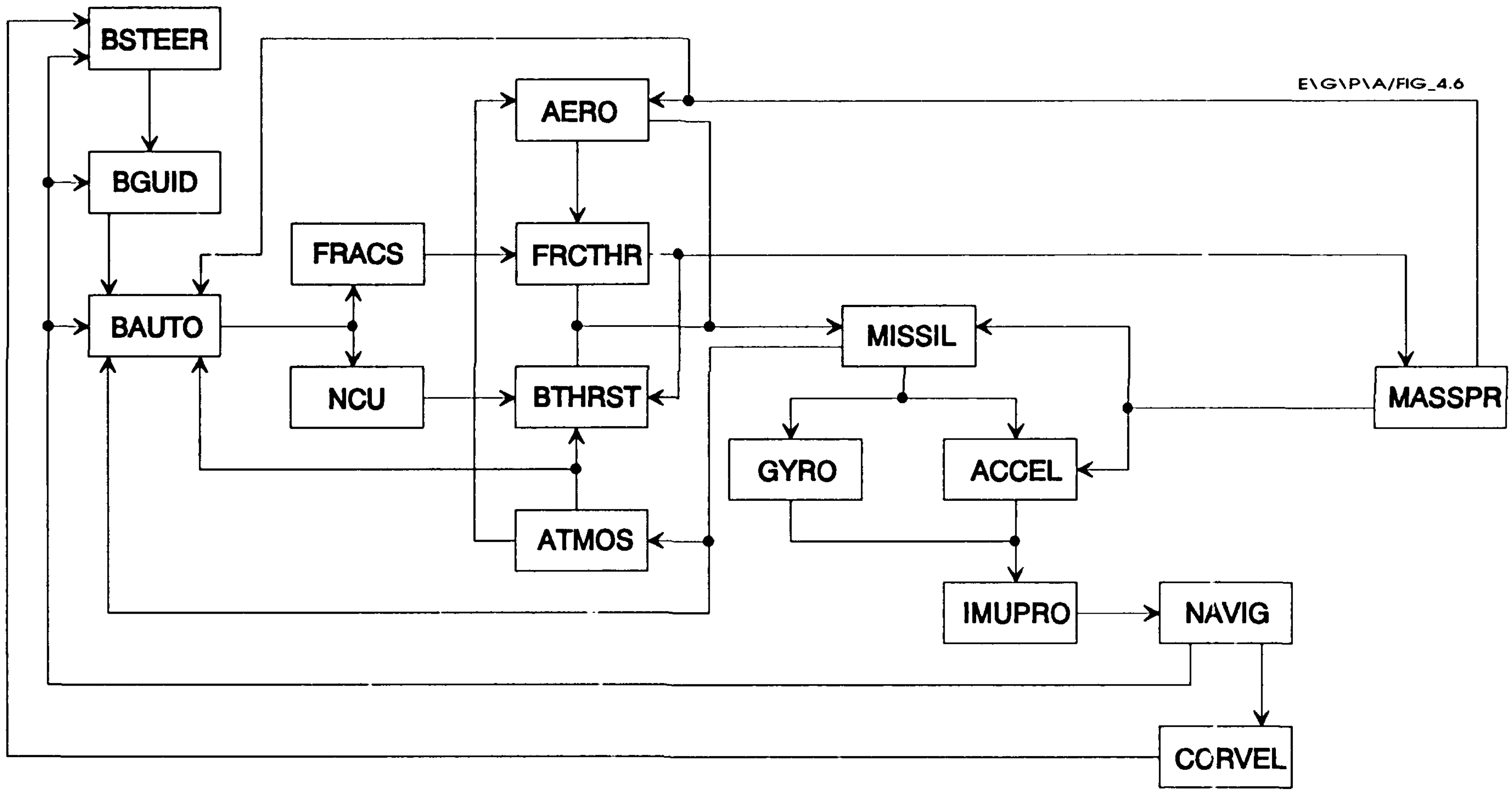


Figure 4.6 Exosim Boost Phase Block Diagram

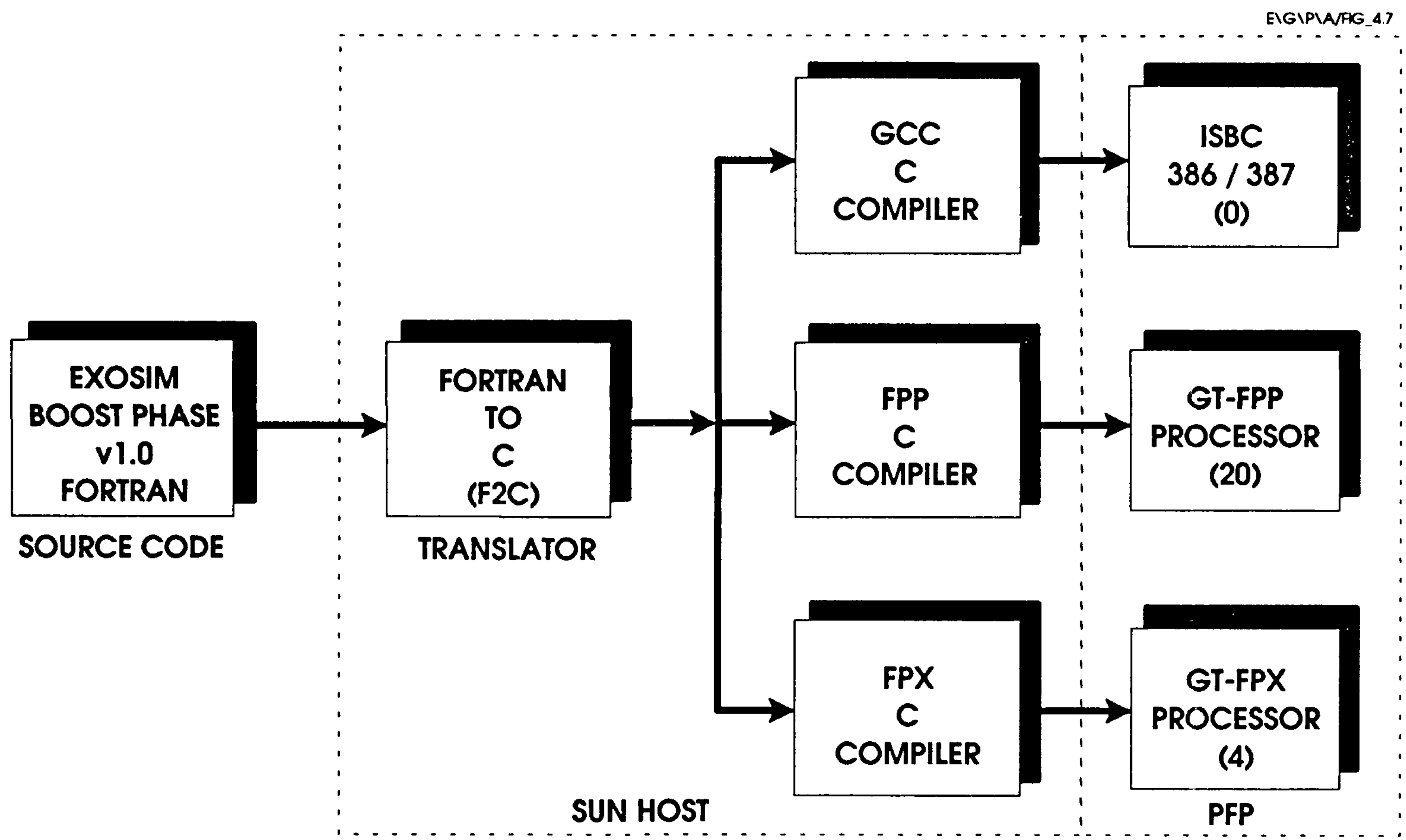


Figure 4.7 Parallel Exosim Boost Phase: Fortran Version

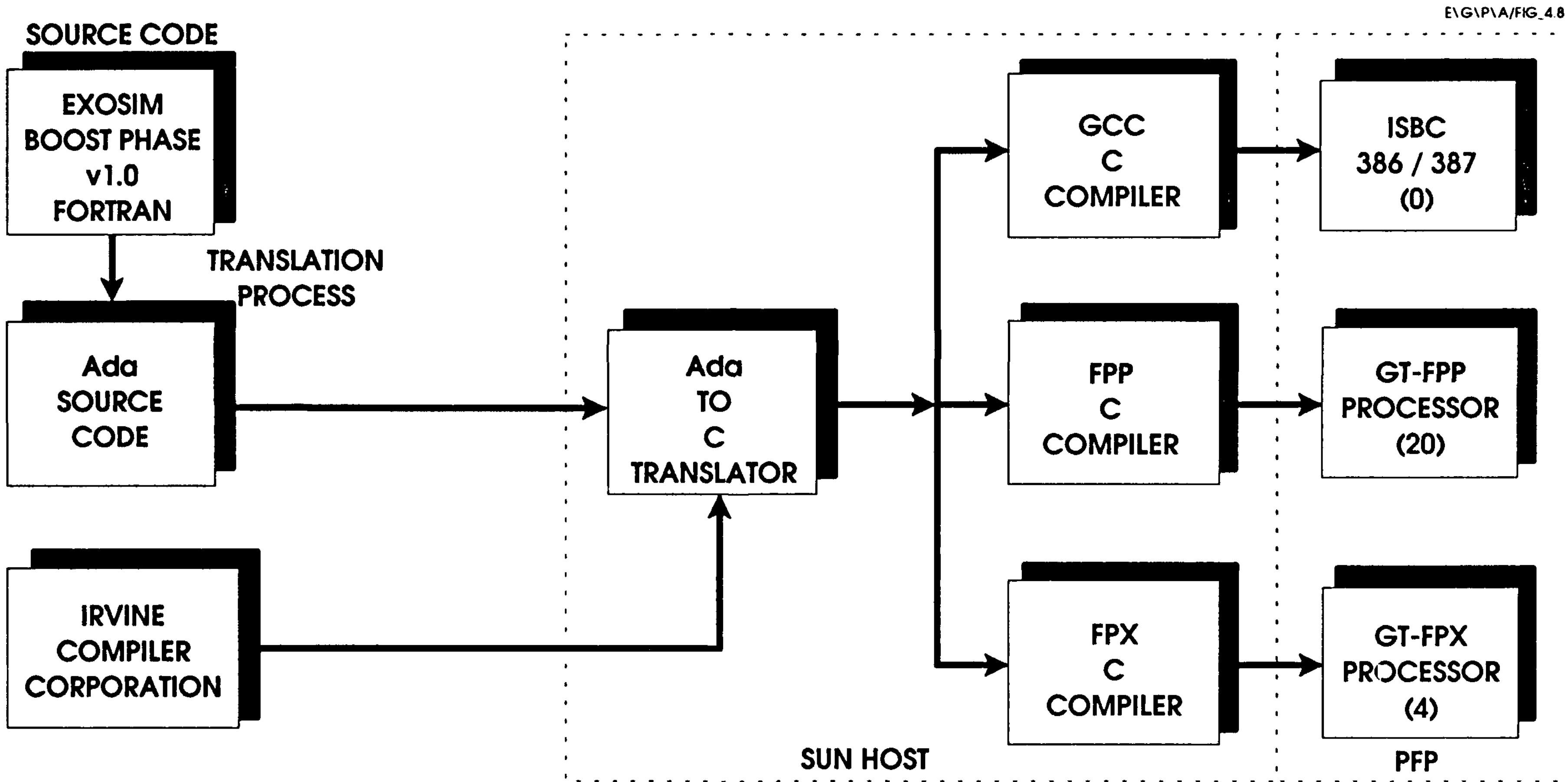


Figure 4.8 Parallel Exosim Boost Phase: Ada Version

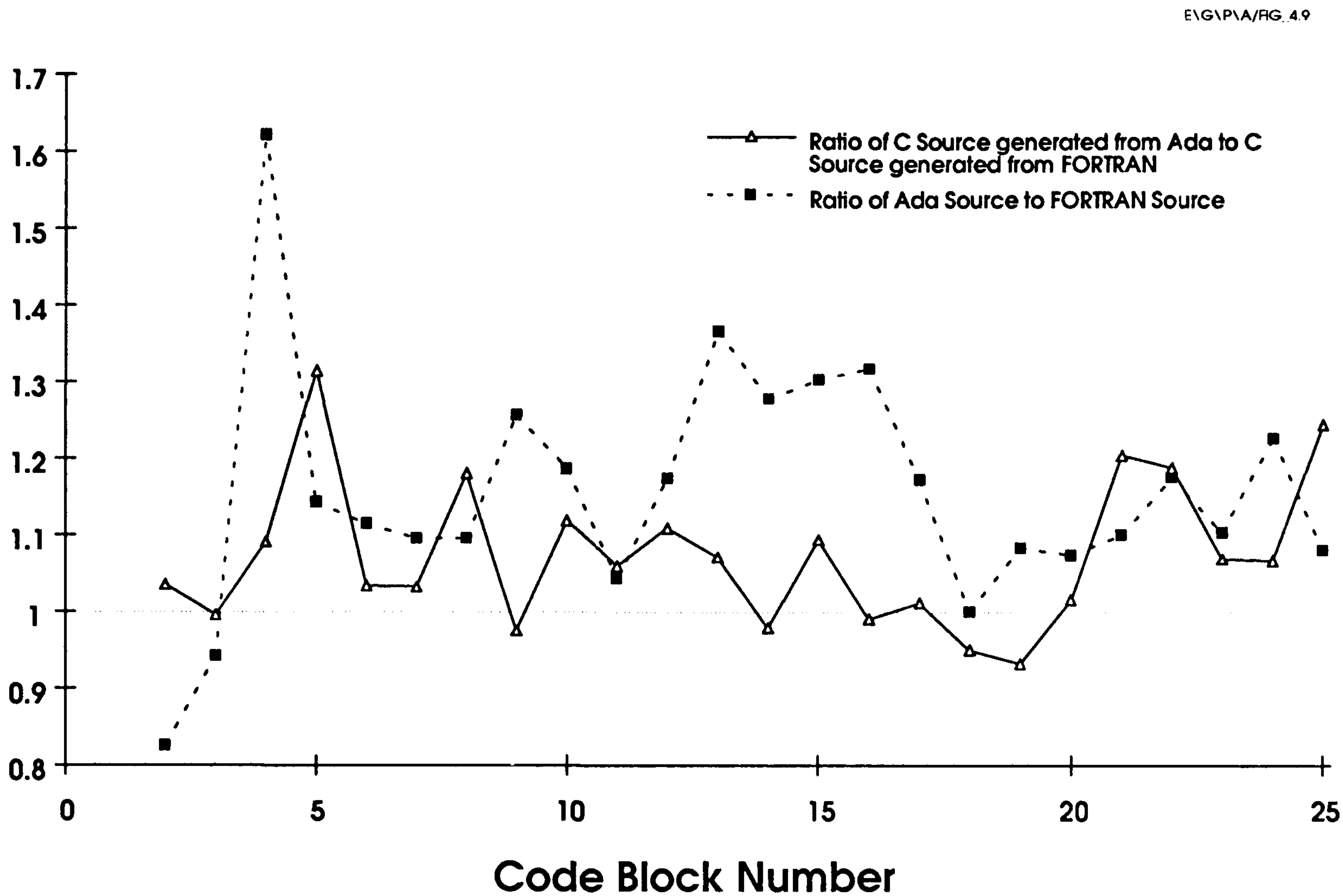


Figure 4.9 Comparison of Source Code and Intermediate C Code

E\G\P\A/FIG_4.10

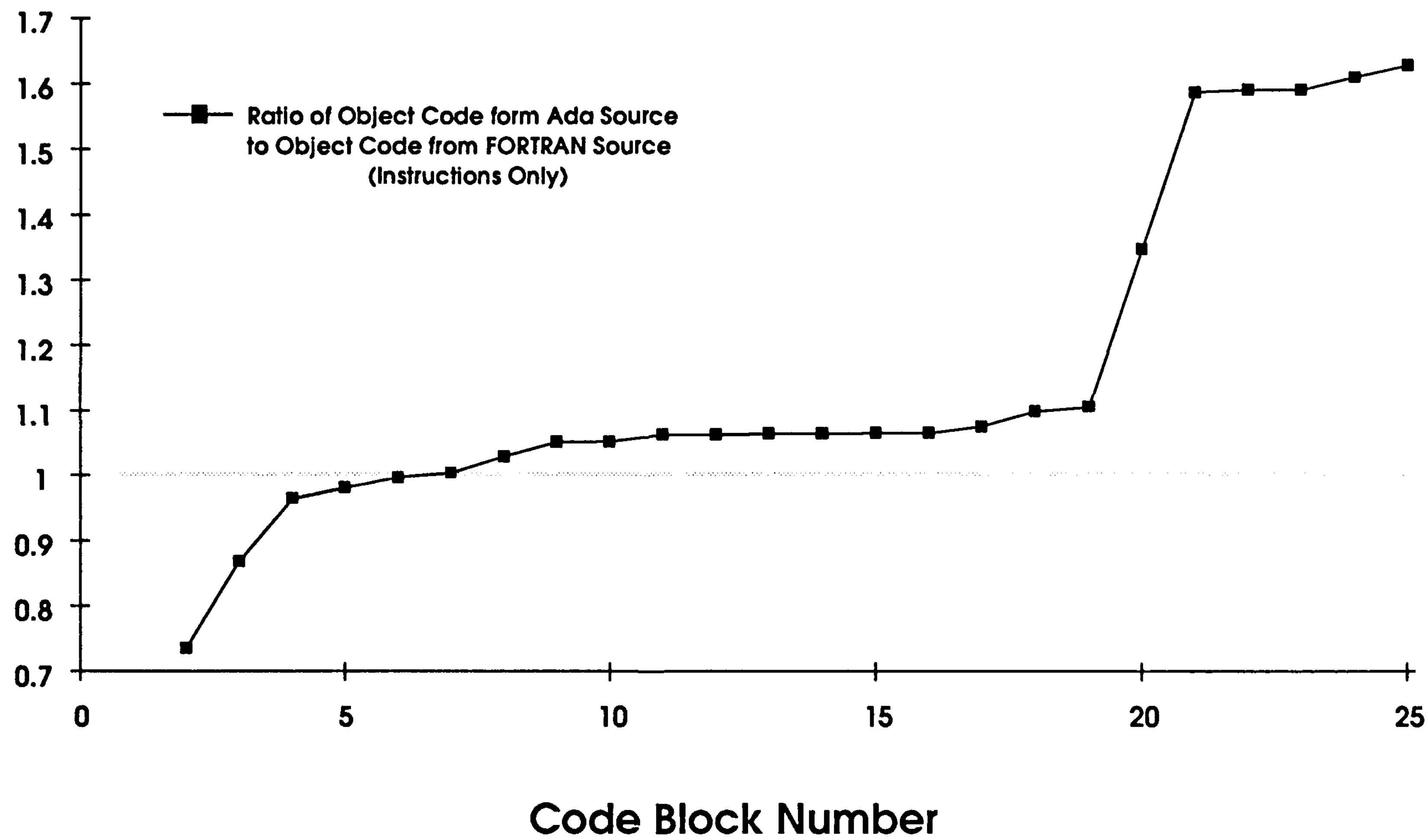


Figure 4.10 Comparison of Object Code Size

E\G\P\A/FIG_4.11

PROCESSOR TYPE	NO. PROCESSORS	RELATIVE EXECUTION TIME
PFP	37	1.0
iSBC 286	1	91.8
MICROVAX II	1	36.3
iSBC 386	1	20.9
T800	1	19.2

Figure 4.11 Benchmark Execution Time For Exosim End-To-End

5.0 References

1. A.M. Henshaw, S.R. Gieseking, R.W. Melton, *Signal Processing Algorithms: Georgia Tech Benchmarks*, Special Technical Report No. STR-0142-90-008, CERL, Georgia Tech, Atlanta, Ga., February 27, 1990.

2. Rani Indaheng, *Signal Processing Algorithms: Georgia Tech Ada Benchmark*, Special Technical Report No STR-0142-91-014, CERL, Georgia Tech, Atlanta, Ga., August 5, 1991.

3. A. Register and A. Henshaw, *Staggered Row Focal Plane Array Analysis*, Special Technical Report No. STR-0142-91-013, CERL, Georgia Tech, Atlanta, Ga., May 1, 1991.

4. "EXOSIM Version 1.0 Simulation Description, Volume 1, " Interim Technical Report, U.S. Army Strategic Defense Command, Contract No. DASG60-88-C-0002, Coleman Research Corporation, 30, June 1989.

5. "EXOSIM Version 1.0 Simulation Description, Volume 2," Interim Technical Report, U.S. Army Strategic Defense Command, Contract No. DASG60-88-C-0002, Coleman Research Corporation, 30, June 1989.

6. "EXOSIM Version 2.0 Simulation Description, Volume 1," Interim Technical Report, CHR/89-2176, U.S. Army Strategic Defense Command, Contract No. DASG60-88-C-0002, Coleman Research Corporation, 30 October 1989.

7. "EXOSIM Version 2.0 Simulation Description, Volume 2," Interim Technical Report, CHR/89-2176, U.S. Army Strategic Defense Command, Contract No. DASG60-88-C-0002, Coleman Research Corporation, 30 October 1989.